

Utility Patent Application Preliminary Amendment

December 21, 2001

Honorable Commissioner of Patents and Trademarks
Washington, DC 20231

Dear Sir:

Accompanying this letter is a preliminary amendment for:

- (a) An application for U.S. Patent by applicants Amber D. Greenwalt, Randy J. Ellis, Adam S. Benz, Robert G. Utterback, Christopher D. Atkins, Deborah L. Maznaritz, Timothy P. Brown, Bryan J. McGlade, and Zhejing (nmn) Luo for NETWORK ELEMENT TERMINAL DATA INTERFACE SYSTEM AND METHOD.
- (b) Replacement pages for the Specification in the above-captioned Patent Application.
- (c) Replacement claims for Claims 48 and 60 in the above-captioned Patent Application.
- (d) Replacement drawing for FIG. 19 in the above-captioned Patent Application.

Respectfully submitted,

 12/21/2001

Kevin Mark Klughart, Ph.D., P.E., J.D., M.I.P., L.L.M.

USPTO Registration Number 39252

5068 W Plano Pkwy, STE 300, Plano, TX 75093

tel: (972) 738-8505

fax: (972) 381-4201

10025076-135863

Preliminary Amendment

Invention: NETWORK ELEMENT TERMINAL DATA INTERFACE SYSTEM
AND METHOD

Inventors: Amber D. Greenwalt, Randy J. Ellis, Adam S. Benz, Robert G. Utterback,
Christopher D. Atkins, Deborah L. Maznaritz, Timothy P. Brown, Bryan
J. McGlade, and Zhejing (nmn) Luo

Filing Date: 12/21/2001

Docket: 135863/ATL-2001-008

Exp Mail: ET702669256US

Introduction

The above-named applicants hereby respectfully request that the following clerical errors be corrected in the above-captioned patent application attached herewith.

In the Specification

(Corrected pages are attached for Examiner use).

On page 15 in line 5, change

--- illustrates a exemplary ---

to

--- illustrates exemplary ---.

On page 34 in line 9, change

--- and/or ---

to

--- and ---.

On page 35 in line 13, change

--- generalize ---

to

--- generalized ---.

On page 36 in line 7, change

--- develop ---

to

--- development ---.

On page 40 in line 1, change

--- (10) ---

to

--- (9) ---.

On page 44 in line 2, change

--- Command ---

to

--- Response ---.

In the Claims

(Corrected pages are attached for Examiner use).

On page 71, resequence claim steps (11) to (20) to read as
(1) through (10) in claim 48.

On page 71, line 19, change

--- (10) ---

to

--- (9) ---.

On page 74, resequence claim steps (21) to (25) to read as
(1) through (5) in claim 60.

In the Drawings

(Corrected pages are attached for Examiner use).

In FIG. 19, change Adapter Database callout

--- 0508 ---

to

--- 1951 ---.

Request for Reconsideration

Applicants request that the instant Patent Application be reconsidered for allowance after the above clerical errors have been corrected. Please contact the representing attorney below if any additional assistance is needed in affecting these clerical errors.

Respectfully submitted,

 12/21/2001

Kevin Mark Klughart, Ph.D., P.E., J.D., M.I.P., L.L.M.

USPTO Registration Number 39252

5068 W Plano Pkwy, STE 300, Plano, TX 75093

tel: (972) 738-8505

fax: (972) 381-4201

10026076-135104
TOTAL 9/28/2001

FIG. 12 illustrates an exemplary drop-down list from the dialog box of FIG. 11;

FIG. 13 illustrates an expanded exemplary view of the dialog box of FIG. 12;

5 FIG. 14 illustrates exemplary Parameter Help windows that are extracted from XML files and their associated I/O manuals;

FIG. 15 illustrates an exemplary network element login dialog box;

10 FIG. 16 illustrates an exemplary annotated TL1DAT operator interface dialog box;

FIG. 17 illustrates an exemplary operator/user interface structure useful in some preferred embodiments of the present invention;

15 FIG. 18 illustrates an exemplary flowchart showing a preferred embodiment of a method used to automatically generate an adapter database from network element source documentation;

20 FIG. 19 illustrates an exemplary data flow of a generic method of producing terminal data interface modules from a source XML network element I/O manual as taught by the present invention;

FIG. 8 (0800) illustrates an exemplary XML file excerpt that is the result of running the HTML file of FIG. 7 (0700) through the **IOManual.pl** PERL script (0604). Generally, one skilled in the art will recognize that the following must be supplied to these PERL scripts to affect the desired translation:

1. NE product name (e.g., 1631SX),
2. NE stream name (e.g., LMC-APS),
3. NE release/version number (e.g., R10.00), and
4. filenames to process.

One skilled in the art will recognize that other parameter combinations are possible. Items 1-3 are what are retrieved from the **RTRV-PRMTR-NE** command at login and are provided to the PERL script by TL1DAT; the location of the files to process are specified by the user when installing the NE (location of the I/O Manual). The name of the resulting XML file is generally a concatenation of items 1-3. There is more content within an actual XML file (DTD information at the top, header-type elements, etc.), which will be well known to one skilled in the art.

Generalized Translation Method (1900, 2000)

The translation of NE I/O manuals to corresponding adapter databases used to construct network element terminal interfaces is a specific example of the more generalized method taught by the present invention which involves the use of XML files as a source repository for all information necessary to describe a network element or some other equipment function. From this XML description, the present invention teaches that a wide variety of documentation and software interfaces can be readily generated automatically given the right filtering paradigm.

This generalized process in terms of data flow is illustrated in FIG. 19 (1900). In this generalized paradigm, a user (1910) who is typically a network element specialist, uses some data interface (1920) to generate an XML file (1930) that is the essential I/O manual for the network element. This I/O manual file (1930) may have several components (chapters, sections, etc.) that contain specific information related to NE specific features such as TL1 syntax (1931), on-line help (1932), or any other type of use information specific or generic to a particular network element.

By the use of any number of filters (1940) that can be implemented in a wide variety of ways, the I/O manual file

(1930) can be filtered to extract information necessary to construct the normal I/O manual reference (1950), adapter database (1951), on-line help subsystem (1952), and any number of other functions and/or features that are normally associated with a software interface necessary to support the network element.

This approach greatly reduces the development and maintenance effort associated with network element support and terminal interface software by permitting a generic XML file (1930) to incorporate all information associated with and important to the network element. The I/O manual (1930) can then be expanded at will to incorporate new network element features without a need to redeploy software to all existing network element installations. This data-driven approach also drastically reduces the complexity of terminal data interface software associated with network elements, as the complexity of command syntax and associated parameter variations is buried within the XML I/O manual file (1930) rather than being incorporated in terminal interface software.

Thus, the present invention teaches the use of an XML I/O manual file (1930) to fully define the feature set and command syntax of a given network element and the subsequent use of filters (1940) to extract necessary information in

(10) repeating steps (1)-(9) as necessary for each network element type and/or version (0910).

While one skilled in the art will recognize that there are equivalent variants to this method, the gist of the disclosed invention permits updating the network element terminal data interface without the need for laborious and time consuming software distributions and/or installations that characterize the prior art. The disclosed method not only permits command trees to be automatically updated, it also permits the end-user tutorials and help files to be automatically updated as well, permitting a tighter integration of maintenance and support functions as compared to the prior art update methods.

Exemplary TL1DAT Operator/User Interface (1000, 1600, 1700)

The TL1DAT Operator/User Interface is divided into four main parts: menubar (1610), command window (1620), response window (1630), and the command creation panel (1640). Exemplary implementations of this user interface are illustrated in FIG. 10 (1000) (unannotated) and FIG. 16 (1600) (annotated). An abstraction of this concept is illustrated in FIG. 17 (1700).

Response Window (1630)

The Response Panel (1630) displays all of the responses received from the network element. The responses are displayed in a scroll pane so that the user can view past responses by scrolling up.

Command Creation Panel (1100, 1640)

The Command Creation Panel (1100, 1640) is made up of three parts: Command Code Tree (1641), Command Parameters Panel (1642), and the Command String Panel (1643). All the information needed for these panels is retrieved from the XML files comprising the adapter database.

- **Command Code Tree (1641)** - Displays all of the command codes, for the NE that is currently connected, either in alphabetical or functional order. Note that the Command Code Tree (1641) example provided in FIG. 16 is illustrated using alphabetical order and that other forms of group and/or functional order are possible. This would, for example, permit related commands (or commands that are typically sequenced) to be displayed for easy selection in the Command Code Tree (1641).
- **Command Parameters Panel (1642)** - This panel provides an editable entry field for each parameter available in this command. If a list of valid values is available for the

48. A NETWORK ELEMENT TERMINAL DATA INTERFACE (NETDI)
software update method comprising:

(1) distributing an I/O manual storage medium to a
customer;

5 (2) distributing a TL1DAT application storage medium
to said customer;

(3) installing said TL1DAT with an on-line user guide;

(4) loading a network element I/O manual from said I/O
manual storage medium;

10 (5) copying said network element I/O manual to a local
storage medium;

(6) selecting a network installation option in said
TL1DAT application;

15 (7) specifying a target location for said I/O manual
to said TL1DAT application;

(8) creating an XML file from said I/O manual files;

(9) using said XML files to create enhanced dialog
screens for presentation to said customer; and

20 (10) repeating steps (1)-(9) as necessary for each
network element type and/or version.

60. A NETWORK ELEMENT TERMINAL DATA INTERFACE (NETDI)
adapter database method comprising:

(1) generating I/O source documents for a network
element;

5 (2) converting said I/O source documents to a standard
file format and optionally inserting hidden tags
if necessary to flag specific adapter database
data;

10 (3) extracting command information from said standard
file format to generate an XML file for an adapter
database;

(4) extracting a command function menu from said
standard file format to generate an XML File for
said adapter database; and

15 (5) repeating steps (1)-(4) for each network element
type and version.

61. The NETWORK ELEMENT TERMINAL DATA INTERFACE (NETDI)
method of Claim 60 wherein one or more steps of said
method is implemented using PERL scripts.

20 62. The NETWORK ELEMENT TERMINAL DATA INTERFACE (NETDI)
method of Claim 60 wherein one or more steps of said
method is implemented on a personal computer (PC).

FIG. 19

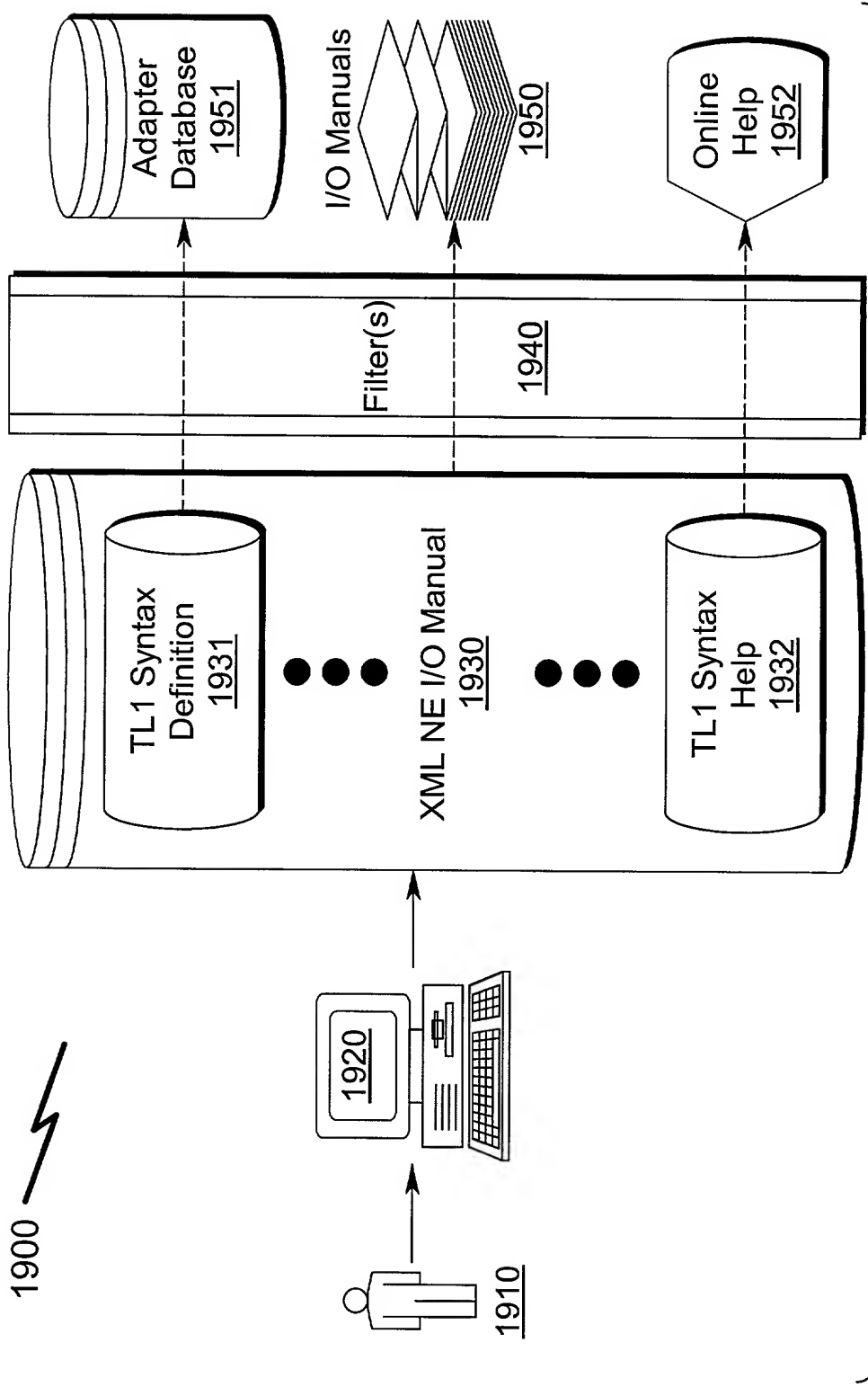


FIG. 20
(2000)